

Trabalho de Extensão

Matéria: Inteligência Artificial



Dayane Teodoro

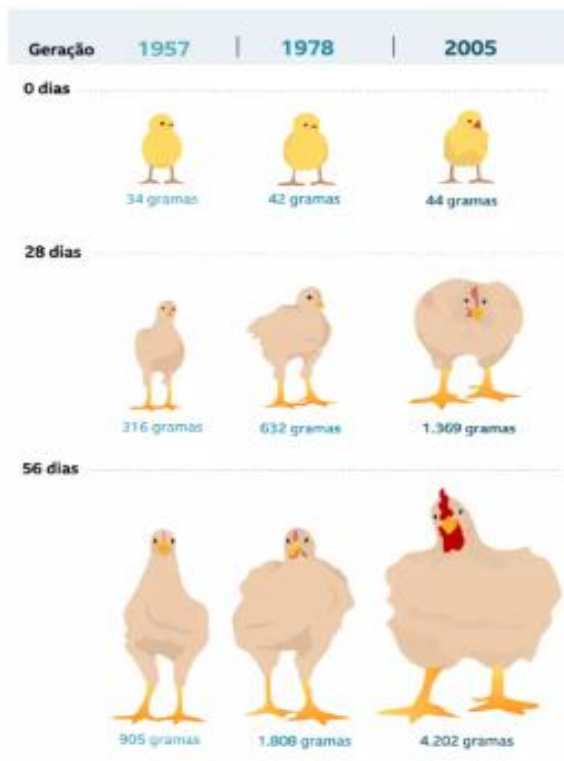
Introdução

Falando um pouco sobre o tema proposto para o desenvolvimento desse trabalho, bem como as tecnologias envolvidas e seus resultados:

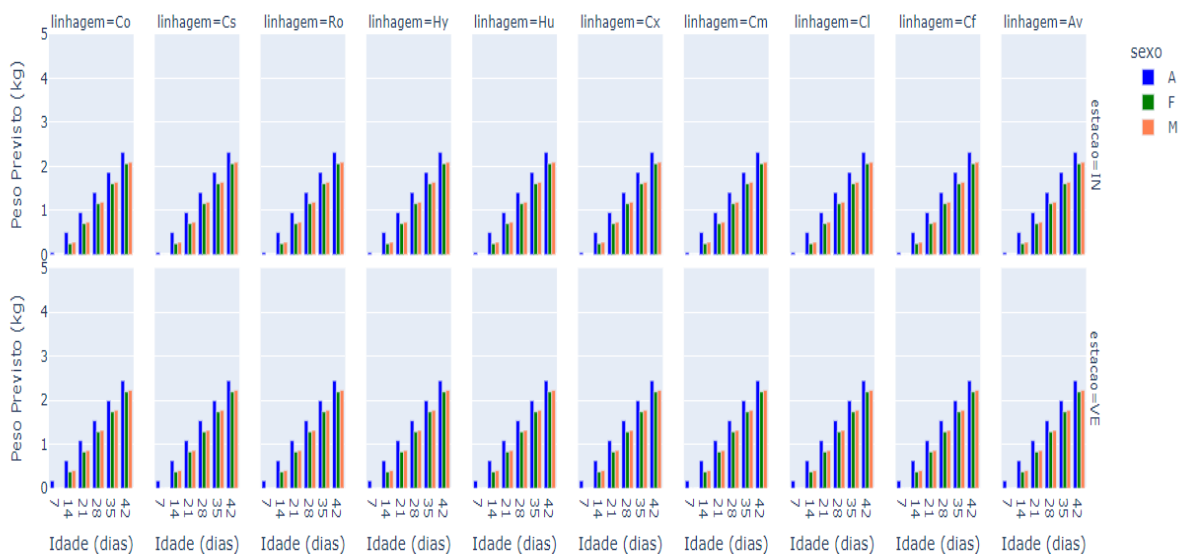
Para este trabalho, utilizei o modelo preditivo de regressão múltipla com aprendizado de máquina para prever o peso dos pintinhos na idade final.

Sobre a evolução dos pintinhos nos últimos anos

Em pouco mais de meio século, o tamanho médio de um frango de corte aumentou 400%. A tecnologia ajudou no crescimento dos frangos principalmente através de avanços genéticos, permitindo a seleção de aves com características que favorecem o rápido ganho de peso e maior produção de carne. Além disso, melhorias na nutrição, com rações mais balanceadas e ricas em nutrientes, potencializaram o crescimento. Tecnologias de manejo em granjas, como controle de temperatura e iluminação, também criaram ambientes mais propícios para o desenvolvimento rápido e saudável das aves.

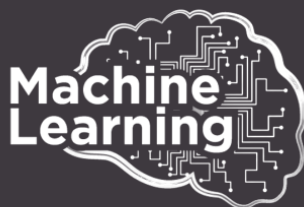


Previsão do Peso Médio por Sexo, Linhagem, Estação e Idade





Explicando o código utilizado
para prever o peso médio dos
pintinho na idade final.



Passo 1: Carregar os Dados

Antes de começar a construir o modelo, precisamos carregar os dados. O código a seguir mostra como fazer isso diretamente do Google Drive, onde o arquivo CSV está armazenado:

```
Python
1 import pandas as pd
2 from google.colab import drive
3
4 # Conectando ao Google Drive
5 drive.mount('/content/drive')
6
7 # Carregar os dados do CSV
8 dados = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Avecom/Dados_2024-10-03_atualizado.csv', sep=';')
9
10 # Verificar o cabeçalho para entender a estrutura
11 print(dados.head())
12
13
```

Aqui, carregamos os dados e verificamos os primeiros registros para garantir que as colunas estejam corretas. As variáveis independentes são idade, sexo, estação do ano e linhagem, e nossa variável dependente, que queremos prever, é o peso_misto_avg.

Passo 2: Preparar as Variáveis

Em seguida, separamos as variáveis independentes (X) da variável dependente (y). A variável `peso_misto_avg` é convertida para o formato numérico, uma vez que é originalmente uma string com vírgulas.

```
Python
1 X = dados[['idade', 'cod_sexo', 'cod_estacao', 'cod_linhagem']]
2 y = dados['peso_misto_avg'].str.replace(',', '.').astype(float)
3
4
5
```

Passo 3: Dividir os Dados

Agora, dividimos o conjunto de dados em treinamento e teste (80% treino, 20% teste). Isso nos permite avaliar a performance do modelo em dados que ele ainda não viu.

```
Python
1 from sklearn.model_selection import train_test_split
2
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
4
5
6
```

Passo 4: Pré-processamento das Variáveis Categóricas

As variáveis categóricas, como sexo, estação e linhagem, precisam ser convertidas em uma representação numérica para serem usadas no modelo. Usamos o OneHotEncoder para esse trabalho

```
Python

1 from sklearn.preprocessing import OneHotEncoder
2 from sklearn.compose import ColumnTransformer
3
4 categorical_columns = ['cod_sexo', 'cod_estacao', 'cod_linhagem']
5
6 # Aplicar OneHotEncoder nas colunas categóricas
7 column_transformer = ColumnTransformer(
8     transformers=[('encoder', OneHotEncoder(drop='first', handle_unknown='ignore'), categorical_columns)],
9     remainder='passthrough'
10 )
11
```


Passo 5: Criando e Treinando o Modelo

Agora, criamos o modelo de regressão linear e o integramos em um pipeline, que primeiro transforma as colunas e, em seguida, aplica a regressão.

```
Python

1 from sklearn.pipeline import Pipeline
2 from sklearn.linear_model import LinearRegression
3
4 modelo = Pipeline(steps=[
5     ('transformer', column_transformer),
6     ('regressor', LinearRegression())
7 ])
8
9 # Treinar o modelo
10 modelo.fit(X_train, y_train)
11
12
13
14
```

Passo 6: Avaliação do Modelo

Após o treinamento, fazemos previsões e avaliamos o desempenho do modelo usando o Erro Médio Quadrático (MSE). Quanto menor o MSE, melhor o modelo.

```
Python
1 from sklearn.metrics import mean_squared_error
2
3 y_pred = modelo.predict(X_test)
4 precisao = mean_squared_error(y_test, y_pred)
5 print(f"Erro médio quadrático (MSE): {precisao:.2f}")
6
7
8
9
10
```

Passo 7: Visualização dos Resultados

Este passo cria um gráfico de colunas interativo para visualizar as previsões de peso com base nas variáveis linhagem, sexo, estação e idade. O gráfico organiza as linhagens em facetas de colunas e as estações em linhas, enquanto as cores representam os diferentes sexos (A, F, M) usando tons claros e escuros. O eixo X mostra a idade dos pintinhos (7, 14, 21, 28, 35, 42 dias) e o eixo Y o peso previsto, limitado entre 0 e 5 kg.

```
Python
1 cores_sexo = {
2     'A': 'blue',
3     'F': 'green',
4     'M': 'coral'
5 }
6
7 fig_coluna = px.bar(
8     df_resultados,
9     x='idade',
10    y='peso_previsto',
11    color='sexo',
12    barmode='group',
13    facet_col='linhagem',
14    facet_row='estacao',
15    title='Previsão do Peso Médio por Sexo, Linhagem, Estação e Idade',
16    labels={'peso_previsto': 'Peso Previsto (kg)', 'idade': 'Idade (dias)'}
17 )
18
19 fig_coluna.for_each_trace(
20     lambda t: t.update(marker_color=cores_sexo[t.name])
21 )
22
23 fig_coluna.update_yaxes(range=[0, 5])
24
25 fig_coluna.update_xaxes(tickvals=[7, 14, 21, 28, 35, 42])
26
27 fig_coluna.show()
```

Passo 8: Previsão para Diferentes Cenários

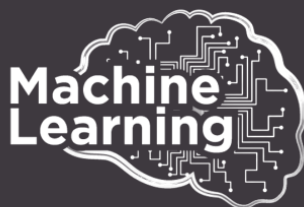
Por fim, utilizamos o modelo treinado para fazer previsões de peso na idade de 42 dias, considerando diferentes combinações de sexo, linhagem e estação..

```
Python
1 sexos = ['A', 'F', 'M']
2 linhagens = ['Co', 'Cs', 'Ro', 'Hy', 'Hu', 'Cx', 'Cm', 'CL', 'CF', 'Av']
3 estacoes = ['IN', 'VE']
4
5 resultados = []
6 for sexo in sexos:
7     for linhagem in linhagens:
8         for estacao in estacoes:
9             novo_X = pd.DataFrame({
10                 'idade': [42],
11                 'cod_sexo': [sexo],
12                 'cod_estacao': [estacao],
13                 'cod_linhagem': [linhagem]
14             })
15             peso_previsto = modelo.predict(novo_X)[0]
16             resultados.append({'sexo': sexo, 'linhagem': linhagem, 'estacao': estacao, 'peso_previsto': peso_previsto})
17
18 df_resultados = pd.DataFrame(resultados)
19 print(df_resultados)
20
21 for index, row in df_resultados.iterrows():
22     print(f"Previsão para Sexo: {row['sexo']}, Linhagem: {row['linhagem']}, Estação: {row['estacao']}: {row['peso_previsto']:.2f} kg na idade de 42 dias")
23
```

Com esse modelo preditivo de regressão múltipla, conseguimos prever o peso dos pintinhos com base em múltiplos fatores.



Conclusão



Conclusão

A regressão múltipla é uma técnica fundamental em análise de dados para prever uma variável com base em múltiplos fatores.

O uso da regressão múltipla para prever o peso médio dos pintinhos baseado em variáveis como idade, sexo, estação e linhagem mostrou-se eficaz. Com técnicas como o pré-processamento de dados categóricos e a divisão em treino e teste, conseguimos construir um modelo preditivo robusto. Esse modelo permite prever com precisão o peso dos pintinhos em diferentes cenários, auxiliando na tomada de decisões em granjas e no gerenciamento do crescimento dos animais de forma otimizada.

Obrigada por ler até aqui

Para mais informações, acesse meu GitHub:



<https://github.com/Dayanebiaerafa>